

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of:

KI-DONG LEE, ET AL.

Application No.:

Filed:

For: **Apparatus and Method for Dynamic
Resource Allocation In Interactive
Satellite Multimedia System**

Art Group:

Examiner:

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REQUEST FOR PRIORITY

Sir:

Applicant respectfully requests a convention priority for the above-captioned application, namely:

<u>COUNTRY</u>	<u>APPLICATION NUMBER</u>	<u>DATE OF FILING</u>
Korea	10-2002-0078827	11 December 2002

☒ A certified copy of the document is being submitted herewith.

Respectfully submitted,

Blakely, Sokoloff, Taylor & Zafman LLP

Dated: 7/22/03

12400 Wilshire Blvd., 7th Floor
Los Angeles, California 90025
Telephone: (310) 207-3800


Eric S. Hyman, Reg. No. 30,139

대한민국 특허청

KOREAN INTELLECTUAL PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto
is a true copy from the records of the Korean Intellectual
Property Office.

출원번호 : 10-2002-0078827
Application Number

출원년월일 : 2002년 12월 11일
Date of Application DEC 11, 2002

출원인 : 한국전자통신연구원
Applicant(s) Electronics and Telecommunications Research Institute



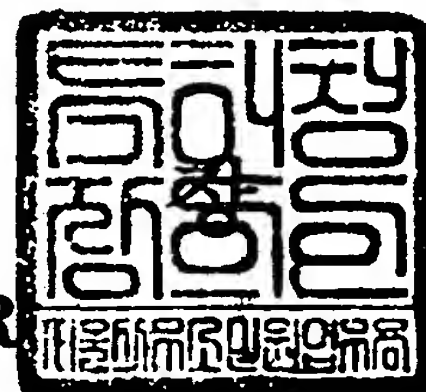
2003 년 05 월 09 일

특

허

청

COMMISSIONER



【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0003
【제출일자】	2002.12.11
【발명의 명칭】	양방향 위성 멀티미디어 시스템에서의 동적 자원 할당 장치 및 그 방법
【발명의 영문명칭】	Apparatus and method for dynamic resource allocation in interactive satellite multimedia system
【출원인】	
【명칭】	한국전자통신연구원
【출원인코드】	3-1998-007763-8
【대리인】	
【명칭】	특허법인 신성
【대리인코드】	9-2000-100004-8
【지정된변리사】	변리사 정지원, 변리사 원석희, 변리사 박해천
【포괄위임등록번호】	2000-051975-8
【발명자】	
【성명의 국문표기】	이기동
【성명의 영문표기】	LEE, Ki Dong
【주민등록번호】	721123-1233012
【우편번호】	122-011
【주소】	서울특별시 은평구 응암1동 36 녹번빌라 202
【국적】	KR
【발명자】	
【성명의 국문표기】	조용훈
【성명의 영문표기】	CHO, Yong Hoon
【주민등록번호】	630622-1000211
【우편번호】	305-755
【주소】	대전광역시 유성구 어은동 한빛아파트 122-1204
【국적】	KR

【발명자】**【성명의 국문표기】**

이호진

【성명의 영문표기】

LEE, Ho Jin

【주민등록번호】

580912-1845717

【우편번호】

305-761

【주소】

대전광역시 유성구 전민동 464-1 엑스포아파트 212-801

【국적】

KR

【심사청구】

청구

【취지】

특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인
특허법인 신성 (인)

【수수료】**【기본출원료】**

20 면 29,000 원

【가산출원료】

14 면 14,000 원

【우선권주장료】

0 건 0 원

【심사청구료】

11 항 461,000 원

【합계】

504,000 원

【감면사유】

정부출연연구기관

【감면후 수수료】

252,000 원

【기술이전】**【기술양도】**

희망

【실시권 허여】

희망

【기술지도】

희망

【첨부서류】

1. 요약서·명세서(도면)_1통

【요약서】**【요약】****1. 청구범위에 기재된 발명이 속한 기술분야**

본 발명은 양방향 위성 멀티미디어 시스템에서의 동적 자원 할당 장치 및 그 방법과, 상기 방법을 실현시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체에 관한 것임.

2. 발명이 해결하려고 하는 기술적 과제

본 발명은, 양방향(대화형) 위성 멀티미디어 시스템에서 많은 수의 단말국과 서로 다른 자원 요청량에 대해 처리량을 최대로 하고, 자원 할당을 위한 할당계획 문제를 빠른 시간안에 만들어 내기 위한 동적 자원 할당 장치 및 그 방법과, 상기 방법을 실현시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체를 제공하고자 함.

3. 발명의 해결방법의 요지

본 발명은, 양방향 위성 멀티미디어 시스템에서의 자원 할당 장치에 있어서, 슈퍼프레임 기간동안 발생하는 자원 요청량을 각 단말국에 대해 누적하기 위한 자원 요청량 수집수단; 각 단말국들에 대해 누적된 자원 요청량을 하나의 슈퍼프레임내에 포함된 프레임씩으로 나누어, 그 몫과 나머지의 반올림한 값의 합을 요청량으로 기억하기 위한 자원 요청량 처리수단; 및 상기 자원 요청량 처리수단에서의 요청량을 기반으로 최적의 할당량을 결정하여, 최적의 할당량을 기반으로 프레임씩에 대하여 각 단말국에게 할당할 타임슬롯들을 결정하기 위한 자원 할당수단을 포함함.

4. 발명의 중요한 용도

1020020078827

출력 일자: 2003/5/10

본 발명은 양방향 위성 멀티미디어 시스템 등에 이용됨.

【대표도】

도 2

【색인어】

리턴링크, 슈퍼프레임, 자원 할당, 프레임짜, 위성

【명세서】**【발명의 명칭】**

양방향 위성 멀티미디어 시스템에서의 동적 자원 할당 장치 및 그 방법{Apparatus and method for dynamic resource allocation in interactive satellite multimedia system}

【도면의 간단한 설명】

도 1 은 본 발명이 적용되는 양방향 위성 멀티미디어 시스템의 구성 예시도.

도 2 는 본 발명에 따른 동적 자원 할당 장치의 일실시에 구성도.

도 3 은 본 발명에 따른 동적 자원 할당 방법에 대한 일실시에 흐름도.

도 4 는 본 발명에 이용되는 슈퍼프레임의 일예시도.

도 5a 내지 5c 는 본 발명의 실시예에 따라 실험 결과치를 나타낸 설명도.

* 도면의 주요 부분에 대한 부호의 설명

21 : 자원 요청량 수집부

22 : 자원 요청량 처리부

23 : 자원 할당부

24 : 자원 할당량 결정부

25 : 자원 할당계획 수립부

26 : 자원 할당계획 복사부

【발명의 상세한 설명】**【발명의 목적】****【발명이 속하는 기술분야 및 그 분야의 종래기술】**

- <10> 본 발명은 양방향(대화형) 위성 멀티미디어 시스템에서 효율적인 주파수 자원 할당 기술 분야에 관한 것으로서, 보다 상세하게는 다중 주파수 시분할 다중 접속(MF-TDMA : Multi-frequency Time Division Multiple Access)을 기반으로 한 리턴링크(단말국에서 위성으로 향하는 링크)의 타임슬롯을 최대의 처리량(throughput)을 갖도록 동적으로 운용할 수 있는 동적 자원 할당 장치 및 그 방법과, 상기 방법을 실현시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체에 관한 것이다.
- <11> 여기서, 대상으로 하는 주파수 자원은 단말국(RCST)에서 위성으로 향하는 리턴링크(return link)에 사용되는 주파수대역, 구체적으로 MF-TDMA를 기반으로 하는 리턴링크에서의 유한한 개수의 타임슬롯의 묶음이 된다. 그리고, MF-TDMA에서의 자원 할당은 프레임(frame) 또는 슈퍼프레임(superframe) 단위로 할당계획이 이루어지는데, 하나의 프레임은 여러 개의 타임슬롯들로 구성되고, 하나의 슈퍼프레임은 여러 개의 프레임들로 구성된다.
- <12> 위성망에서 TDMA의 자원 할당에 관한 종래의 기술로서, 위성간 회선에서의 자원할당(timeslot assignment in inter-satellite links) 방법, 이동통신 시스템에서 사용되는 TDMA 방식에 대한 자원할당 방법 등이 있다.
- <13> 그런데, 양방향(대화형) 위성 멀티미디어 시스템은 여러 개의 단말국들(RCST)이 위성을 통하여 중심국(Hub)에 서비스 요청을 하고 중심국(Hub)이 이에 응하여 위성을 통하여 단

말국(RCST)에게 서비스를 제공하는 위성 네트워크이다. 위성 통신 시스템은 주파수의 상대 가치가 지상 이동통신 시스템에 비하여 매우 높으므로, 주어진 주파수 자원에 대해 사용자 트래픽에 대한 처리량(throughput)을 최대화하는 것이 무엇보다 중요하다. 즉, 위성 통신 시스템에서의 주파수 자원은 지상 이동통신망의 주파수 자원의 가격보다 매우 비싸다. 따라서, 사용자 트래픽 전송을 위한 처리량을 최대화하는 문제는 통신의 단위비용을 낮추고 사업이윤의 극대화로 직결되므로 매우 중요하다.

<14> 특히, 양방향(대화형) 위성통신 시스템에서는 단말국(RCST)에 의해 이루어지는 자원 요청에서 중심국(Hub)에 의해 처리되는 자원의 할당계획, 그리고 이 계획을 단말국(RCST)이 다시 받기까지의 전송지연(round-trip time)을 고려하여 자원이 할당되어야 하는데, 이러한 점에서 기존에 연구 개발된 위성 관련 자원할당 기술과는 큰 차이가 있다.

<15> 뿐만 아니라, 이동통신 시스템에서의 타임슬롯 할당계획/패킷 할당계획(timeslot scheduling/packet scheduling)은 상기한 바와 같은 전송지연에서 차이점이 있는 시스템 상에서의 문제이며, 사용하는 주파수의 대역폭(bandwidth)이 위성이 상대적으로 높으며, 전송지연으로 인하여 시시각각 변화하는 트래픽에 적응하기 어려우므로 프레임의 길이도 길게 설계되는 차이가 있다.

<16> 결과적으로, 스케줄링할 타임슬롯의 개수가 크게 증가하게 되며, 이에 따라 타임슬롯 할당을 위한 스케줄링의 복잡도가 매우 증가하게 된다.

<17> 따라서, 양방향(대화형) 위성 멀티미디어 시스템에서는 여러 개의 단말국(RCST)이 생성하는 주파수 자원 요구를 중심국(Hub)이 수신하여 동적으로 운용되는 주파수 자원 할당의 계획을 매우 신속하게 출력할 수 있는 방안이 필수적으로 요구된다.

【발명이 이루고자 하는 기술적 과제】

<18> 본 발명은, 상기한 바와 같은 요구에 부응하기 위하여 제안된 것으로, 양방향(대화형) 위성 멀티미디어 시스템에서 많은 수의 단말국과 서로 다른 자원 요청량에 대해 처리량을 최대화 하고, 자원 할당을 위한 할당계획 문제를 빠른 시간안에 만들어 내기 위한 동적 자원 할당 장치 및 그 방법과, 상기 방법을 실현시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체를 제공하는데 그 목적이 있다.

【발명의 구성 및 작용】

<19> 상기 목적을 달성하기 위한 본 발명은, 양방향 위성 멀티미디어 시스템에서의 자원 할당 장치에 있어서, 슈퍼프레임 기간동안 발생하는 자원 요청량을 각 단말국에 대해 누적하기 위한 자원 요청량 수집수단; 각 단말국들에 대해 누적된 자원 요청량을 하나의 슈퍼프레임내에 포함된 프레임씩으로 나누어, 그 몫과 나머지의 반올림한 값의 합을 자원 요청량으로 기억하기 위한 자원 요청량 처리수단; 및 상기 자원 요청량 처리수단에서의 자원 요청량을 기반으로 최적의 할당량을 결정하여, 최적의 할당량을 기반으로 프레임씩에 대하여 각 단말국에게 할당할 타임슬롯들을 결정하기 위한 자원 할당수단을 포함하여 이루어진 것을 특징으로 한다.

- <20> 그리고, 본 발명은 양방향 위성 멀티미디어 시스템에서의 자원 할당 방법에 있어서, 수퍼프레임 기간동안 발생하는 자원 요청량을 각 단말국에 대해 누적하는 제 1 단계; 각 단말국들에 대해 누적된 자원 요청량을 하나의 수퍼프레임내에 포함된 프레임 짝으로 나누어, 그 몫과 나머지의 반올림한 값의 합을 요청량으로 기억하는 제 2 단계; 및 상기 제 2 단계의 요청량을 기반으로 최적의 할당량을 결정하여, 최적의 할당량을 기반으로 프레임짝에 대하여 각 단말국에게 할당할 타임슬롯들을 결정하는 제 3 단계를 포함하여 이루어진 것을 특징으로 한다.
- <21> 한편, 본 발명은 프로세서를 구비한 자원 할당 장치에, 수퍼프레임 기간동안 발생하는 자원 요청량을 각 단말국에 대해 누적하는 제 1 기능; 각 단말국들에 대해 누적된 자원 요청량을 하나의 수퍼프레임내에 포함된 프레임짝으로 나누어, 그 몫과 나머지의 반올림한 값의 합을 요청량으로 기억하는 제 2 기능; 및 상기 제 2 기능에서의 요청량을 기반으로 최적의 할당량을 결정하여, 최적의 할당량을 기반으로 프레임짝에 대하여 각 단말국에게 할당할 타임슬롯들을 결정하는 제 3 기능을 실현시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체를 제공한다.
- <22> 본 발명은 MF-TDMA 시스템에서 자원(타임슬롯) 할당을 효율적으로 함으로써 처리량을 극대화하고자 한다. 즉, MF-TDMA에서의 자원 할당은 프레임(frame) 또는 수퍼프레임(superframe) 단위로 할당계획이 이루어지므로, 본 발명은 많은 수의 단말국과 서로 다른 자원 요청량에 대해 처리량(throughput)을 최대로 하고, 이 할당계획 문제를 가능한 한 빠른 시간안에 만들어내고자 한다.
- <23> 이를 위해, 본 발명은 단말국에서 위성으로 향하는 링크에서의 처리량을 높이기 위한 동적으로 자원(타임슬롯) 할당계획을 출력하는 알고리즘을 제안한다.

- <24> 본 알고리즘은 문제의 분할(problem decomposition) 방식의 접근방법을 활용하여 계산량을 대폭 줄였으며, 각 하부 문제에서는 신속한 최적의 할당량 결정, 그리고 최적 할당량을 기반으로 신속하게 타임슬롯 할당하는 방법을 제안하였다. 이는 출력되는 해의 최적성이 보장되며, 처리시간이 매우 빠르므로 처리량 극대화를 이루면서 동시에 트래픽의 동적변화에 매우 빠르게 적응하는 장점을 가진 방법이다.
- <25> 상술한 목적, 특징들 및 장점은 첨부된 도면과 관련한 다음의 상세한 설명을 통하여 보다 분명해 질 것이다. 이하, 첨부된 도면을 참조하여 본 발명에 따른 바람직한 일 실시예를 상세히 설명한다.
- <26> 도 1 은 본 발명이 적용되는 양방향 위성 멀티미디어 시스템의 구성 예시도이다.
- <27> 전술한 바와 같이, 양방향(대화형) 위성 멀티미디어 시스템은 여러 개의 단말국들(RCST)(11)이 위성(12)을 통하여 중심국(Hub)(13)에 서비스 요청을 하고, 중심국(Hub)(13)이 이에 응하여 위성(12)을 통해 단말국(RCST)(11)에게 서비스를 제공하는 위성 네트워크이다. 여기에서는 주어진 주파수 자원에 대해 사용자 트래픽에 대한 처리량(throughput)을 최대화하는 것이 무엇보다 중요하다.
- <28> 상기 양방향(대화형) 위성 멀티미디어 시스템은 도 1에 도시된 바와 같이 하나의 위성(12), 하나의 중심국(Hub)(13), 그리고 여러 개의 단말국(RCST)(11)으로 이루어진다. 다만, 둘 이상의 위성, 둘 이상의 중심국(Hub), 그리고 여러 개의 단말국(RCST)으로 시스템을 구성하고자 하는 경우에는 본 시스템 형태를 둘 이상 조합함으로써 가능하다.
- <29> 여러 개의 단말국(11)은 각자 필요한 자원을 중심국(13)에 요청하며, 이에 대해 중심국(13)이 허락하는 자원(타임슬롯)만을 사용한다.

- <30> 중심국(13)은 단말국들(11)로부터 들어온 자원 요청을 취합하여, 각 단말국(11)에게 할당할 타임슬롯을 결정하여 위성(12)을 통해 단말국(11)에게 보고한다.
- <31> 특히, 중심국(13)에는 단말국(11)에서 위성(12)으로 향하는 링크에서의 처리량을 높이기 위한 동적으로 자원(타임슬롯) 할당계획을 출력하는 알고리즘이 탑재되어 있어, 중심국(13)은 여러 개의 단말국(11)이 생성하는 주파수 자원 요구를 수신하여, 동적으로 운용되는 주파수 자원 할당의 계획을 매우 신속하게 출력한다. 이때, 대상으로 하는 주파수 자원은 단말국(11)에서 위성(12)으로 향하는 리턴링크에 사용되는 주파수대역, 즉 MF-TDMA를 기반으로 하는 리턴링크에서의 유한한 개수의 타임슬롯의 묶음이 된다. 이러한 자원 할당계획을 매 수퍼프레임마다 신속하게 출력하기 위하여, 원 문제(original problem)(하기의 [수학식 1] 및 [수학식 3] 참조)를 두 가지의 하부 문제, 즉 각 단말국별 할당량 결정문제(문제A)와 결정된 할당량을 기반으로 각 단말국에게 할당할 타임슬롯들을 선별하는 문제(문제B)로 재구성하여 풀이속도를 대폭 개선하며, 문제A를 매우 빨리 풀기 위한 알고리즘(하기의 [수학식 4] 참조) 및 문제B를 매우 빨리 풀기 위한 알고리즘(하기의 [수학식 5] 참조)을 제안한다.
- <32> 이러한 문제분할(problem decomposition) 방법을 통하여 계산량 축소효과 및 계산 속도 개선효과가 발생한다.
- <33> 그럼, 도 2 및 도 3을 참조하여 중심국(13)에서의 동적 자원 할당 장치(실시간 자원 관리기) 및 그 방법을 보다 상세하게 살펴보기로 한다.
- <34> 도 2 는 본 발명에 따른 동적 자원 할당 장치의 일실시에 구성도이다.

- <35> 중심국(Hub)(13)에 있는 동적 자원 할당 장치(실시간 자원 관리기)는 자원 요청량 수집부(21), 수집된 자원 요청량을 프레임쌍의 개수로 나누어 반올림 적용하여 몫을 취하는 자원 요청량 처리부(22), 수퍼프레임내의 첫 번째 프레임쌍에서의 자원 할당량 결정부(24) 및 자원 할당계획 수립부(25), 그리고 동일 수퍼프레임내의 다른 프레임쌍에 대한 자원 할당계획 복사부(25)로 구성된다.
- <36> 즉, 첫째 프레임쌍에 대해 할당계획을 수립함에 있어서, 자원 할당량 결정부(24)는 최적 할당량을 결정하되, 첫째 프레임쌍에 대해 각 단말국(RCST)(11)에게 얼마 만큼씩 할당할 것인가를 결정하고, 자원 할당계획 수립부(25)는 최적 할당량을 기반으로 할당계획을 수립하되, 결정된 할당량을 가지고 어떤 슬롯을 어느 단말국(RCST)(11)에게 할당할 것인가를 결정한다. 그리고, 자원 할당계획 복사부(26)는 동일 수퍼프레임내의 나머지 프레임쌍에 첫째 프레임쌍의 할당계획을 그대로 복사한다.
- <37> 이를 도 2를 참조하여 보다 구체적으로 살펴보면, 본 발명에 따른 중심국(13)의 동적 자원 할당 장치는, 수퍼프레임 기간동안 발생하는 자원 요청량을 각 단말국(RCST)(11)에 대해 누적하기 위한 자원 요청량 수집부(21)와, 각 단말국들(RCST)(11)에 대해 누적된 자원 요청량을 하나의 수퍼프레임(superframe)내에 포함된 프레임쌍(frame pair)으로 나누어, 그 몫과 나머지의 반올림한 값의 합을 요청량으로 기억하기 위한 자원 요청량 처리부(22)와, 자원 요청량 처리부(22)에서의 요청량을 기반으로 최적의 할당량을 결정하여, 최적의 할당량을 기반으로 프레임쌍에 대하여 각 단말국(RCST)(11)에게 할당할 타임슬롯들을 결정하기 위한 자원 할당부(23)를 포함한다.
- <38> 상기 자원 할당부(23)는 자원 요청량 처리부(22)에서 처리된 요청량을 기반으로 최적의 할당량을 결정하기 위한 자원 할당량 결정부(24)와, 최적의 할당량을 기반으로 첫

번째 프레임쪽에 포함된 가용 타임슬롯을 어느 단말국(RCST)(11)에게 할당할 것인지를 결정하기 위한 자원 할당계획 수립부(25)와, 첫 번째 프레임쪽의 타임슬롯 할당계획을 동일 슈퍼프레임내의 나머지 프레임쪽에 복사하기 위한 자원 할당계획 복사부(26)를 포함한다.

<39> 자원 요청량 수집부(21)는 슈퍼프레임 기간동안 발생하는 자원 요청량을 각각의 단말국(RCST)(11)에 대해 누적하여 자원 요청량 처리부(22)에게 보고한다. 보고후, 0으로 재설정되고, 다음 슈퍼프레임이 시작하면 발생하는 자원 요청량에 대해 다시 누적을 시작한다.

<40> 자원 요청량 처리부(22)는 입력된 각 단말국들(RCST)(11)에 대한 자원요청량을 하나의 슈퍼프레임내에 포함된 프레임쪽의 개수로 나누어 소수점 이하를 반올림하며, 자원 할당량 결정부(24)는 이들을 감점상수(비용, 가중치 등의 의미로 할당량 결정시 우선순위를 부여함)에 따라 최적의 할당량을 결정한다(하기의 [수학식 4] 참조).

<41> 프레임쪽에서의 자원 할당계획 수립부(25)는 결정된 최적 할당량을 기반으로 첫 번째 프레임쪽(슈퍼프레임의 시간상 가장 빠른 것이 첫 번째)에 포함된 모든 가용 타임슬롯을 어느 단말국(RCST)(11)에게 할당할 것인가를 결정한다(하기의 [수학식 5] 참조).

<42> 동일 슈퍼프레임내의 다른 프레임쪽에 대한 자원 할당계획 복사부(26)는 첫 번째 프레임쪽의 타임슬롯 할당계획을 동일 슈퍼프레임내의 나머지 프레임쪽에 그대로 복사하는 기능을 담당한다.

<43> 도 3 은 본 발명에 따른 동적 자원 할당 방법에 대한 일실시에 흐름도이다.

- <44> 먼저, 할당계획을 수립하기 위한 자원 할당 단위를 정의한다(301). 즉, 수퍼프레임(superframe), 프레임짝(frame pair) 등에 대해 정의한다.
- <45> MF-TDMA에서의 자원 할당은 프레임(frame) 또는 수퍼프레임(superframe) 단위로 할당계획이 이루어지는데, 하나의 프레임은 여러 개의 타임슬롯들로 구성되고, 하나의 수퍼프레임은 여러 개의 프레임들로 구성된다.
- <46> 프레임짝은 수퍼프레임의 부분집합으로서, 다음의 조건을 만족한다.
- <47> 하나의 프레임짝은 동일 시간대를 갖는 프레임 혹은 프레임들의 집합이고, 각각의 프레임짝은 서로 다른 프레임짝과 시간대만 다를 뿐 주파수대역 및 타임슬롯의 배치형태가 모두 같다. 따라서, 하나의 수퍼프레임에 존재하는 모든 프레임짝은 서로 소인 집합이고, 이들 모두의 합집합이 수퍼프레임을 이룬다.
- <48> 예를 들면, 도 4에 도시된 바와 같이 32개의 프레임(frame)으로 구성된 수퍼프레임(superframe)의 경우, 프레임0과 프레임16을 하나의 프레임짝(framepair)으로 묶고 나머지 프레임들도 두 개씩 쌍(pair)으로 하여 16개의 프레임짝(framepair)을 구성한다.
- <49> 이후, 자원 요청량 수집부(21)에서 로그인(logon)한 단말국(RCST)(11)의 아이디(ID)들을 기억하고(302), 이들로부터 수퍼프레임 길이동안 누적된 요청량을 벡터형태로 기억한다(303). 이때, 요청량은 갱신(update)된다. 각 단말에 대한 누적된 요청량은 자원 요청량 처리부(22)에서 수퍼프레임이 포함하고 있는 프레임짝으로 나누어 그 몫과 나머지의 반올림한 값의 합을 요청량으로 기억한다. 이는 프레임짝 만큼 각 요청량을 반복하여 할당할 것이기 때문이다. 이때, 작은 오차는 무시된다.

- <50> 다음으로, 자원 할당량 결정부(23)에서 요청량을 기반으로 최적의 할당량을 구하며(304)(하기의 [수학식 4] 참조), 자원 할당계획 수립부(25) 및 자원 할당계획 복사부(26)에서 최적의 할당량을 기반으로 프레임짝(frame pair)에 대하여 각 단말국(RCST)(11)에게 할당할 타임슬롯들을 결정한다(305,306).
- <51> 이때, 특히 자원 할당계획 수립부(25)에서 결정된 최적 할당량을 기반으로 첫 번째 프레임짝(수퍼프레임의 시간상 가장 빠른 것이 첫 번째)에 포함된 모든 가용 타임슬롯을 어느 단말국(RCST)(11)에게 할당할 것인가를 결정하고(305)(하기의 [수학식 5] 참조), 자원 할당계획 복사부(26)에서 첫 번째 프레임짝의 타임슬롯 할당계획을 동일 수퍼프레임내의 나머지 프레임짝에 그대로 복사한다(306).
- <52> 즉, 도 4와 같은 32개의 프레임으로 구성된 수퍼프레임의 경우, 첫째 프레임짝에 대해서만 하기의 [수학식 4] 및 [수학식 5]의 알고리즘으로 풀고 두 번째부터 16번째 프레임짝에 들어있는 타임슬롯 할당 결정은 첫 번째 프레임짝에서의 결정사항을 그대로 복사하여 사용한다. 이는 문제의 크기(결정할 변수가 많으면 풀기가 매우 어려워짐)를 감소시킴으로써 계산 시간을 단축시킬 수 있는 효과가 있다. 예를 들면, 프레임당 2000개의 타임슬롯이 있다고 가정하면, 첫 번째 프레임짝에 대해 2000×2 (프레임0과 프레임16에 들어있는 자원 갯수), 즉 4000개의 할당계획만 짜고 나머지 프레임짝에 대해서는 단지 복사만 하면 된다. 그러나, 프레임짝 개념 및 동일 수퍼프레임내의 다른 프레임짝에 대한 타임슬롯 할당계획 복사 개념을 이용하지 않는다면, 64000개의 할당계획을 짜야 하므로 상대적으로 비효율적임을 알 수 있다.

- <53> 그럼, 동일 수퍼프레임내의 첫 번째 프레임짜에 대한 최적의 할당량 결정 과정 (304) 및 최적 할당량을 기반으로 신속하게 타임슬롯을 할당하는 과정(305)에 대해 보다 상세하게 살펴보기로 한다.
- <54> 우선, 로그온(logon)한 단말국들(RCST)(11)로부터 수퍼프레임 길이동안 누적된 요청량을 기반으로 최적의 할당량을 구하는 과정(304)을 설명한다.
- <55> 상기 [수학식 1]은 타임슬롯 할당을 위한 원 문제(동적 타임슬롯 할당을 위한 이진 정수계획 최적화 모형도)를 보여준다.

<56>

$$\begin{aligned}
 & (CAP) \\
 & MINIMIZE \quad g(x) = \sum_{j \in R} v_j \sum_{m \in A} \sum_{i \in S_m} (1 - x_{ij}) \\
 & Subject \quad to \\
 & \quad \sum_{m \in A} \sum_{i \in S_m} x_{ij} \leq \min \{Q_j, X_j + Y_j\}, j \in R \\
 & \quad \sum_{m \in A} \sum_{i \in S_m} x_{ij} \geq Y_j, j \in R \\
 & \quad \sum_{j \in R} x_{ij} \leq 1, i \in S_m, m \in A \\
 & \quad x_{ij} \in \{0, 1\}
 \end{aligned}$$

【수학식 1】

- <57> 상기 [수학식 1]에서, i 는 i 번째 타임슬롯, j 는 j 번째 로그온된 단말국을 의미하며, R 은 로그온된 단말국의 ID 집합, A 는 가용한(available) 리턴링크 복조기(return link demodulator)의 ID 집합, S_n 은 가용한 타임슬롯의 ID 집합, Q_j 는 단말국 j 가 할당받을 수 있는 타임슬롯의 최대개수, X_j 는 단말국 j 가 요청한 타임슬롯의 개수(추가 할당량), Y_j 는 단말국 j 에게 최소한 할당되어야 하는 타임슬롯의 개수(기본 할당량), x_{ij} 는 타임슬롯 i 가 단말국 j 에게 할당되면 "1", 아니면 "0"임을 표시하는 이진변수(decision variable)(즉, 의사결정변수), v_j 는 할당 행렬 $[x_{ij}]$ 에게 발생하는 penalty 계수

(penalty 가중치 값)를 의미한다. 이 penalty 계수는 각 RCST가 경험하는 지연(delay) 시간 등에 따라 RCST별로 결정될 수 있다. 즉, 오랫동안 전송이 안되고 대기중인 (delayed) 데이터가 많은 RCST일수록 그 penalty값을 크게 배당함으로써 사용자가 경험하는 불편도의 편차를 해소하는 한편 자원할당의 공정성(fairness)을 달성할 수 있다.

<58> 상기 [수학식 1]을 설명하면, $g(x)$ 는 행렬 $[x_{ij}]$ 가 표시하는 할당에 대한 총 penalty이다. 그리고, 제약 조건으로, 첫째 각 단말국에게 할당되는 타임슬롯 개수는 최대량(수퍼프레임 구조 등 시스템 파라미터에 의해 결정되는 최대할당량) 및 요청량 이하이어야 하고, 둘째 각 단말국에게 할당되는 타임슬롯 개수는 최소할당량(서비스품질, grade-of-service를 만족시키기 위해 정하는 값) 이상이어야 하며, 셋째 각 타임슬롯은 두 개 이상의 단말국에게 할당될 수는 없다.

<59> [수학식 1]은 본 발명에서 제안하는 프레임작 개념을 이용하여 자원할당 문제를 효율적으로 풀기 위하여 다음의 [수학식 2]로 변형된다. 상기 [수학식 1]과 [수학식 2]의 차이는, 할당계획을 만들 대상 자원의 개수가 각각 수퍼프레임 안의 총 슬롯 개수와 프레임작 안의 총 슬롯수로서, 어떠한 풀이 알고리즘을 사용하더라도, 문제 크기가 작은 [수학식 2]의 풀이시간이 복잡도 면에서 훨씬 짧다는 점이다. 수퍼프레임안의 슬롯들 중, 할당계획이 이루어지지 않은 나머지 슬롯들에 대해서는 첫째 프레임작의 할당계획을 단순히 복사하며, 이렇게 하여 훨씬 빠른 시간안에 수퍼프레임안의 슬롯들에 대한 할당계획이 만들어진다.

<60>

$$\begin{aligned}
 & (CAP_1) \\
 & \text{MINIMIZE } g_1(x) = \sum_{j \in R} v_j \sum_{m \in A} \sum_{i \in S_{m,1}} (1 - x_{ij}) \\
 & \text{Subject to} \\
 & \sum_{m \in A} \sum_{i \in S_{m,1}} x_{ij} \leq \min \{Q_j, X_j + Y_j\}, j \in R \\
 & \sum_{m \in A} \sum_{i \in S_{m,1}} x_{ij} \geq Y_j, j \in R \\
 & \sum_{j \in R} x_{ij} \leq 1, i \in S_{m,1}, m \in A \\
 & x_{ij} \in \{0, 1\}
 \end{aligned}$$

【수학식 2】

<61> 여기서, $S_{m,1}$ 은 첫 번째 프레임쪽에 포함된 가용 슬롯들의 집합을 의미한다. 원문제 (CAP)을 n (하나의 슈퍼프레임에 속한 프레임쪽의 개수)개로 분할(문제분할, problem decomposition)하였으므로, 최적해 x 에서 $g(x) = \sum_{i=1}^n g_i(x)$ 가 성립한다.

<62> 이 문제는 하기의 [수학식 3]과 같이 할당량 결정문제를 우선적으로 풀어내는 과정(각 단말국들을 위한 최적 할당량 결정을 위한 정수계획 최적화 모형도)을 밟는다.

<63> 할당량 결정문제에서 v_j 는 단말국 j 가 다른 단말국에 비하여 얼마나 중요한가를 나타내는 상대적 가중치이다. 이 가중치는 운용상에 서비스 품질도에 따라 결정될 수 있다. 예를 들어, 지연이 많이 되었던 단말국인 경우 해당 가중치를 높게 줌으로써 우선적 또는 가급적 많은 자원이 할당되도록 활용될 수 있다.

<64>

$$\begin{aligned}
 & \text{MINIMIZE } f(z) = \sum_{j \in R} v_j z_j \\
 & \text{Subject to} \\
 & z_j \leq \min \{Q_j, X_j + Y_j\}, j \in R \\
 & z_j \geq Y_j, j \in R \\
 & \sum_{j \in R} z_j \leq \sum_{m \in A} S_{m,1} \\
 & z_j \in \mathbb{Z} \cup \{0\}
 \end{aligned}$$

【수학식 3】

- <65> 상기 [수학식 3]에서 결정변수는 z_j 이며, 이는 단말국 j 에게 할당해주는 타임슬롯의 개수(할당량)를 의미한다.
- <66> 상기 [수학식 3]을 설명하면, $f(z)$ 는 벡터 z 가 표시하는 할당량에 대한 총 penalty이다. 그리고, 제약 조건으로, 첫째 각 단말국에게 할당량은 최대량 및 요청량 이하이어야 하고, 둘째 각 단말국에게 할당량은 최소량 이상이어야 하며, 셋째 총 할당량은 가용량(가용 타임슬롯 개수) 이하이어야 한다.
- <67> 할당량 결정 문제의 풀이 알고리즘은 하기의 [수학식 4]와 같다.

<68>

Step 1 (Sort $[v_j]$)
 $k := 1, J_{k-1} := \emptyset$
 FOR($k \leftarrow 1$: $k \leftarrow |R|$) {
 $j_k = \arg \max \{v_j, j \in R - J_{k-1}\}$.
 $J_k := J_{k-1} \cup \{j_k\}$.
 }
Step 2 (Find an optimal y^*)
 $m := \max \{m \mid \sum_{k=1}^m X_{j_k} \leq \sum_{m=1}^M |S_{m,1}| - \sum_{j \in R} Y_j\}$.
 If $m = |R|$, then $y_j^* := \min \{Q_j - Y_j, X_j\}$ for $j \in R$.
 Else,
 $y_j^* := \min \{Q_j - Y_j, X_j\}$ for $j \in J_m$.
 $y_{j_{m+1}}^* := \sum_{m=1}^M |S_{m,1}| - \sum_{j \in R} Y_j$
 $\quad - \sum_{j \in J_m} \min \{Q_j - Y_j, X_j\}$.
 $y_{j_k}^* := 0$ for $k = m+2, \dots$.
Step 3 (Find an optimal z^*)
【수학식 4】 $z^* = y^* + [Y_j]$.

- <69> 상기 [수학식 4]에서, Step1은 v_j 를 크기에 따라 큰 것부터 작은 순서로 정렬하는 과정이다. 여기서, j_k 는 k 번째로 큰 $\text{penalty weight}(v_j)$ 를 갖는 단말국의 ID, J_k 는 $\{j_1, \dots, j_k\}$ 로 구성된 집합이다.

- <70> 그리고, Step2는 추가 할당량의 최적 결정 과정으로, $n := \max\{\dots\}$ 에서 (가용 타임슬롯 개수-기본 할당 개수)가 추가 할당량($= \sum x_{j_k}$)으로, 만약 남은 자원량이 충분하면 추가 요청량은 최대로 할당하고, 한편 남은 자원량이 부족하면 우선순위가 높은 단말국(J_n)들에게는 최대로 할당하고 그 보다 낮은 단말국들에게는 할당되고 남은 부분만큼만 할당해 준다.
- <71> 또한, Step3은 총할당량 결정 과정으로, 각 단말국에 대해 (총할당량 = 기본 할당량+추가 할당량)으로 하여 출력한다.
- <72> 상기 [수학식 4]를 참조하여, 최적 자원 할당량을 결정하는 과정(304)을 살펴보면, 우선 Step1에서 가중치가 높은 것부터 낮은 것으로 정렬(sort)한다. 이때, 가중치가 가장 높은 것이 j_1 이라는 인덱스를 갖고 집합 J_1 에 저장된다. 마찬가지로, 가중치가 k 번째로 높은 것이 j_k 라는 인덱스를 갖고, 집합 J_k 에 $\{j_1, \dots, j_k\}$ 들이 저장된다.
- <73> 이후, Step2에서, 최저 할당량을 제외한 추가적 할당량(y_j^*)을 결정한다.
- <74> 다음으로, Step3에서, 결정된 추가적 할당량에 최적할당량을 더하여 총 할당량(z_j^*)을 구한다. 단, $y^* = [y_j^*]$, $z^* = [z_j^*]$ 이다.
- <75> 한편, [수학식 5]는 최적할당량 벡터 z^* 를 기반으로 각 단말국(RCST)에게 할당할 타임슬롯들을 결정하는 과정(305)이다. 즉, 결정된 할당량 $[z_j^*]$ 를 기반으로 각 단말국에게 할당할 타임슬롯들을 고르는 과정이다.

<76>

```

Step 1 (Initialization)
slot_counter ← 1, x ← 0
Step 2 (Iteration)
FOR(k ← 1; k ≤ K; k++) {
  FOR(i ← slot_counter; i ≤ j_k; i++) {
    xi,j_k ← 1
  }
  slot_counter ← ij_k
}

```

【수학식 5】

<77>

상기 [수학식 5]에서, 슬롯 i 가 단말국 j_k 에게 할당되면 x_{i,j_k} 를 1로 세팅한다. 할당량 개수 만큼 i 를 증가시키면서 1로 세팅하므로 단말 j_k 에게 할당되는 양은 z_j^* 개가 된다.

<78>

[수학식 5]에서, 대상 타임슬롯은 첫 번째 프레임짜에 포함된 모든 타임슬롯이다. 가장 낮은 주파수(캐리어)에서부터 시간적으로 증가하는 방향으로 할당하며, 그 다음 캐리어에 대해서 마찬가지로 반복하여 나간다.

<79>

상기 과정(304,305)에서 생성된 프레임짜의 계획을 나머지 프레임짜들에 대해 복사한다(306). 복사를 통해 슈퍼프레임에 포함된 모든 타임슬롯에 대한 할당계획이 완성된다.

<80>

상기에서, 할당량이 최적이 되는 이유는 본 방법으로 나오는 해에서 어느 방향으로 움직여도 목적식의 값이 개선되지 않기 때문이다.

<81>

본 동적 자원 할당 알고리즘의 효율성(신속성)을 실험적으로 증명하기 위해 컴퓨터 프로그램을 통하여 펜티엄 III 1.0GHz PC에서 실험한 결과를 제시한다(도 5a 내지 5c 참조). 이는 128개의 단말국, 65024개의 타임슬롯, 가중치는 모두 1, 최저 할당량은 프레

임당 6개, 최고 할당량은 프레임당 508개, 임의로 생성된 요청량을 입력 데이터로 하여 두 문제를 푸는데 걸리는 시간을 측정한 것이다.

<82> 측정에 사용된 프로그램은 실제 구현중인 중심국(Hub)(13)의 실시간 자원 관리기(동적 자원 할당 장치)에 이식된 형태로서, 실제 시스템에서의 유니트간 메시징을 위한 자료구조 형태로 자원 할당계획을 표현하는데 걸리는 시간이 포함되었다. 계산 결과에 따르면, 할당량 결정은 최대 약 10ms, 타임슬롯 선택은 최대 약 30ms의 시간이 걸린다. 단말국(RCST)(11)에서 일어난 요청이 중심국(Hub)(13)의 할당계획이 완료되어 다시 단말국(RCST)(11)에게 전달되기까지 약 500ms의 시간이 걸리며(정지궤도 위성의 경우), 수퍼프레임의 길이가 보통 2000~92000ms(DVB-RCS 규격의 경우)임을 감안할 때 매우 빠른 수준의 처리 속도라고 결론지을 수 있다.

<83> 상술한 바와 같은 본 발명의 방법은 프로그램으로 구현되어 컴퓨터로 읽을 수 있는 기록매체(씨디롬, 램, 롬, 플로피 디스크, 하드 디스크, 광자기 디스크 등)에 저장될 수 있다.

<84> 이상에서 설명한 본 발명은 전술한 실시예 및 첨부된 도면에 의해 한정되는 것이 아니고, 본 발명의 기술적 사상을 벗어나지 않는 범위 내에서 여러 가지 치환, 변형 및 변경이 가능하다는 것이 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 있어 명백할 것이다.

【발명의 효과】

<85> 상기한 바와 같은 본 발명은, 양방향 위성 멀티미디어 네트워크에서, MF-TDMA를 기반으로 한 리턴링크(단말에서 위성으로)의 타임슬롯을 할당하는 작업에 있어서, 프레임 짝 도입을 이용한 문제분할 방법을 통해 매우 신속하게 최대의 처리량(throughput)을 갖도록 할당계획을 출력할 수 있는 효과가 있다.

【특허청구범위】**【청구항 1】**

양방향 위성 멀티미디어 시스템에서의 자원 할당 장치에 있어서,
수퍼프레임 기간동안 발생하는 자원 요청량을 각 단말국에 대해 누적하기 위한 자원 요청량 수집수단;
각 단말국들에 대해 누적된 자원 요청량을 하나의 수퍼프레임내에 포함된 프레임씩으로 나누어, 그 몫과 나머지의 반올림한 값의 합을 요청량으로 기억하기 위한 자원 요청량 처리수단; 및
상기 자원 요청량 처리수단에서의 요청량을 기반으로 최적의 할당량을 결정하여, 최적의 할당량을 기반으로 프레임씩에 대하여 각 단말국에게 할당할 타임슬롯들을 결정하기 위한 자원 할당수단
을 포함하는 동적 자원 할당 장치.

【청구항 2】

제 1 항에 있어서,
상기 자원 할당수단은,
첫 번째 프레임씩에 대해 각 단말국에게 얼마 만큼씩 할당할 것인가를 결정하여 최적 할당량을 기반으로 첫 번째 프레임씩에 대해서만 타임슬롯 할당계획을 수립하고, 동일 수퍼프레임내의 다른 프레임씩에 대해서는 첫 번째 프레임씩의 타임슬롯 할당계획을 그대로 복사하는 것을 특징으로 하는 동적 자원 할당 장치.



【청구항 3】

제 1 항에 있어서,

상기 자원 할당수단은,

첫 번째 프레임짝에 대해 각 단말국에게 얼마만큼씩 타임슬롯을 할당할 것인가를 결정하기 위한 자원 할당량 결정수단;

상기 자원 할당량 결정수단에서 결정된 할당량을 기반으로 첫 번째 프레임짝에 포함된 가용 타임슬롯을 어느 단말국에게 할당할 것인지를 결정하기 위한 자원 할당계획 수립수단; 및

첫 번째 프레임짝의 타임슬롯 할당계획을 동일 슈퍼프레임내의 나머지 프레임짝에 복사하기 위한 자원 할당계획 복사수단

을 포함하는 동적 자원 할당 장치.

【청구항 4】

제 1 항 내지 제 3 항 중 어느 한 항에 있어서,

상기 프레임짝은,

수퍼프레임의 부분집합으로서, 하나의 프레임짝은 동일 시간대를 갖는 프레임 혹은 프레임들의 집합이고, 각각의 프레임짝은 서로 다른 프레임짝과 시간대만 다를 뿐 주파수대역 및 타임슬롯의 배치형태가 동일하며, 하나의 수퍼프레임에 존재하는 모든 프레임 짝은 서로 소인 집합이고 이들 모두의 합집합이 수퍼프레임을 이루는 것을 특징으로 하는 동적 자원 할당 장치.

【청구항 5】

제 4 항에 있어서,

상기 자원 요청량 처리수단은,

수집된 자원 요청량을 프레임쌍의 개수로 나누어 몫을 취하되, 입력된 각 단말국들에 대한 자원요청량을 하나의 슈퍼프레임내에 포함된 프레임쌍의 개수로 나누어 소수점 이하를 반올림하는 것을 특징으로 하는 동적 자원 할당 장치.

【청구항 6】

양방향 위성 멀티미디어 시스템에서의 자원 할당 방법에 있어서,

슈퍼프레임 기간동안 발생하는 자원 요청량을 각 단말국에 대해 누적하는 제 1 단계;

각 단말국들에 대해 누적된 자원 요청량을 하나의 슈퍼프레임내에 포함된 프레임쌍으로 나누어, 그 몫과 나머지의 반올림한 값의 합을 요청량으로 기억하는 제 2 단계; 및

상기 제 2 단계의 요청량을 기반으로 최적의 할당량을 결정하여, 최적의 할당량을 기반으로 프레임쌍에 대하여 각 단말국에게 할당할 타임슬롯들을 결정하는 제 3 단계를 포함하는 동적 자원 할당 방법.

【청구항 7】

제 6 항에 있어서,

상기 제 3 단계는,

상기 제 2 단계의 요청량을 기반으로 최적의 할당량을 결정하여 최적 할당량을 기반으로 첫 번째 프레임짜에 대해서만 타임슬롯 할당계획을 수립하고, 동일 슈퍼프레임내의 다른 프레임짜에 대해서는 첫 번째 프레임짜의 타임슬롯 할당계획을 그대로 복사하는 것을 특징으로 하는 동적 자원 할당 방법.

【청구항 8】

제 6 항에 있어서,

상기 제 3 단계는,

첫 번째 프레임짜에 대해 각 단말국에게 얼마만큼씩 할당할 것인가를 결정하는 제 4 단계;

최적의 할당량을 기반으로 첫 번째 프레임짜에 포함된 가용 타임슬롯을 어느 단말국에게 할당할 것인지를 결정하는 제 5 단계; 및

첫 번째 프레임짜의 타임슬롯 할당계획을 동일 슈퍼프레임내의 나머지 프레임짜에 복사하는 제 6 단계

를 포함하는 동적 자원 할당 방법.

【청구항 9】

제 8 항에 있어서,

상기 제 4 단계는,

상기 제 2 단계에서 처리된 자원 요청량에 대해 우선순위에 따라 최적의 할당량을 결정하되, 할당량 결정시, 가중치가 높은 것부터 낮은 순으로 정렬하고, 최저 할당량을 제외한 추가적 할당량을 결정하여, 결정된 추가적 할당량에 최적 할당량을 더하여 총 할당량을 구하는 것을 특징으로 하는 동적 자원 할당 방법.

【청구항 10】

제 6 항 내지 제 9 항 중 어느 한 항에 있어서,

상기 제 2 단계는,

수집된 자원 요청량을 프레임쌍의 개수로 나누어 몫을 취하되, 입력된 각 단말국들에 대한 자원 요청량을 하나의 슈퍼프레임내에 포함된 프레임쌍의 개수로 나누어 소수점 이하를 반올림하는 것을 특징으로 하는 동적 자원 할당 방법.

【청구항 11】

프로세서를 구비한 자원 할당 장치에,

슈퍼프레임 기간동안 발생하는 자원 요청량을 각 단말국에 대해 누적하는 제 1 기능;

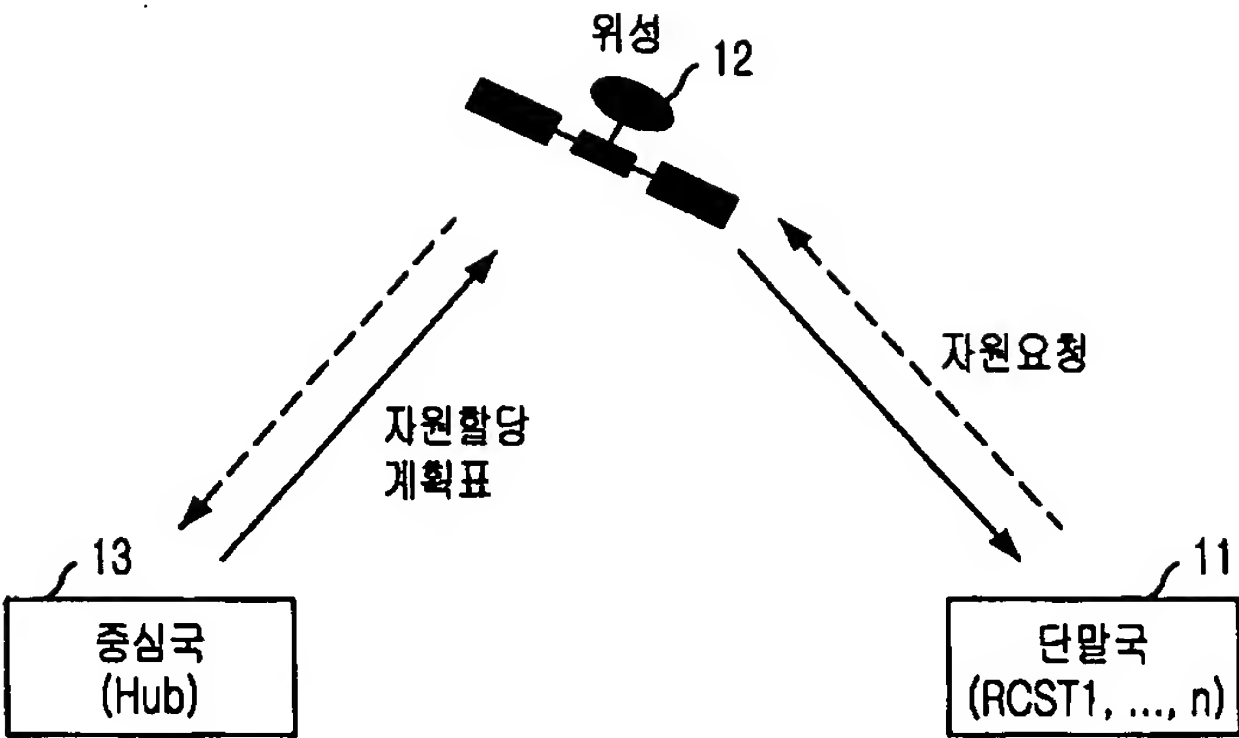
각 단말국들에 대해 누적된 자원 요청량을 하나의 슈퍼프레임내에 포함된 프레임쌍으로 나누어, 그 몫과 나머지의 반올림한 값의 합을 요청량으로 기억하는 제 2 기능; 및

상기 제 2 기능에서의 요청량을 기반으로 최적의 할당량을 결정하여, 최적의 할당량을 기반으로 프레임짜에 대하여 각 단말국에게 할당할 타임슬롯들을 결정하는 제 3 기능

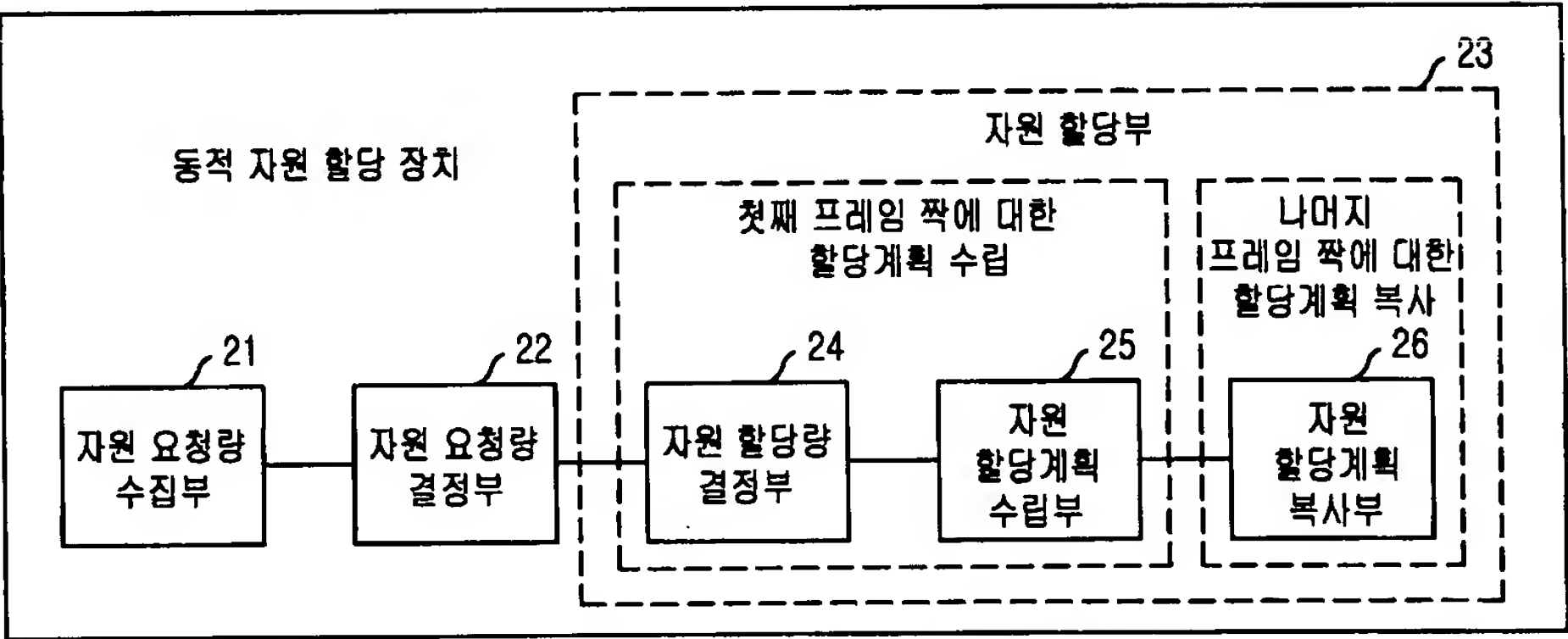
을 실현시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체.

【도면】

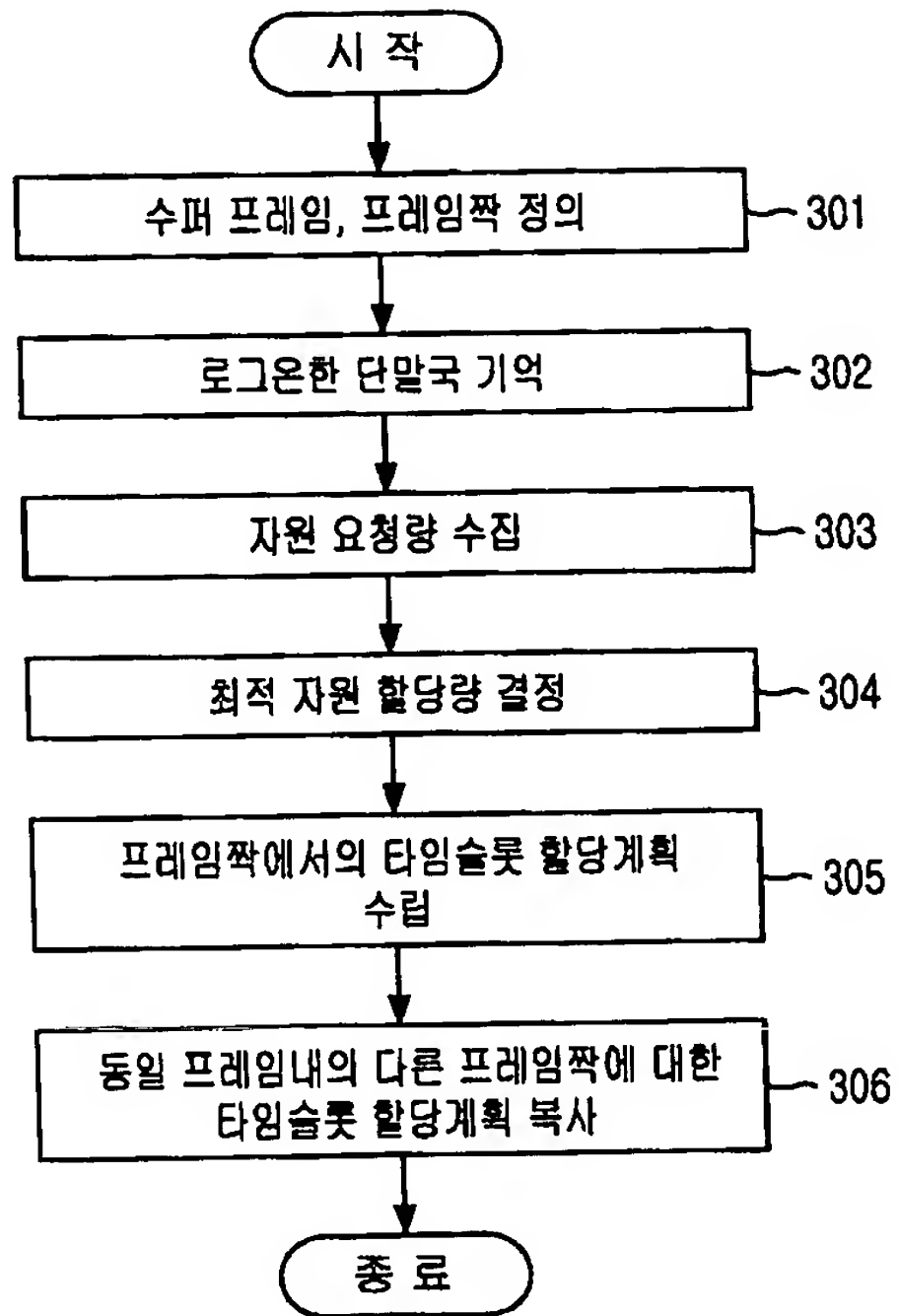
【도 1】



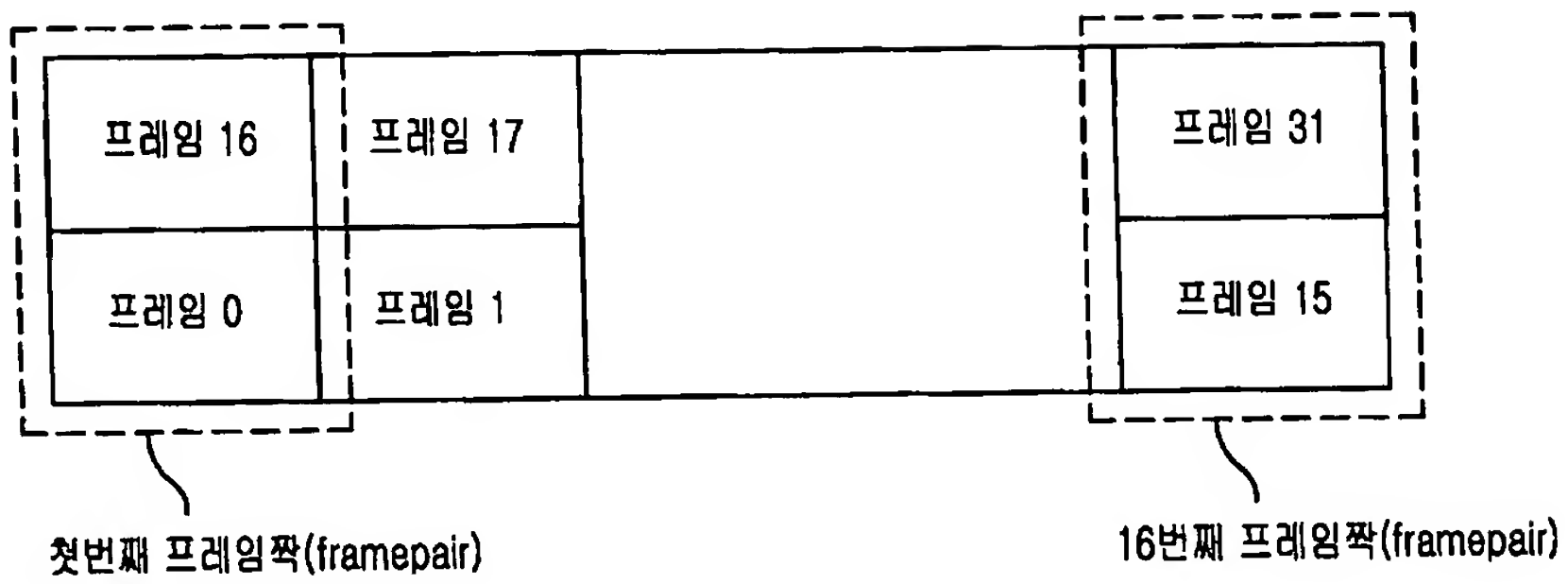
【도 2】



【도 3】



【도 4】



【도 5a】

```

allocation amount
0: 13 1: 14 2: 14 3: 14 4: 15 5: 15 6: 15 7: 15
8: 16 9: 16 10: 16 11: 16 12: 17 13: 17 14: 17 15: 18
16: 18 17: 18 18: 18 19: 19 20: 19 21: 19 22: 20 23: 20
24: 20 25: 20 26: 21 27: 21 28: 21 29: 22 30: 22 31: 22
32: 22 33: 23 34: 23 35: 23 36: 24 37: 24 38: 24 39: 24
40: 25 41: 25 42: 25 43: 25 44: 26 45: 26 46: 26 47: 27
48: 27 49: 27 50: 27 51: 28 52: 28 53: 28 54: 29 55: 29
56: 29 57: 29 58: 30 59: 30 60: 30 61: 31 62: 31 63: 31
64: 31 65: 32 66: 32 67: 32 68: 33 69: 33 70: 33 71: 33
72: 34 73: 34 74: 34 75: 34 76: 35 77: 35 78: 35 79: 36
80: 36 81: 36 82: 36 83: 37 84: 37 85: 37 86: 38 87: 38
88: 38 89: 38 90: 39 91: 39 92: 39 93: 40 94: 40 95: 40
96: 40 97: 41 98: 41 99: 41 100: 42 101: 42 102: 42 103: 42
104: 43 105: 43 106: 43 107: 43 108: 44 109: 44 110: 44 111: 45
112: 45 113: 45 114: 45 115: 46 116: 46 117: 46 118: 47 119: 47
120: 47 121: 47 122: 48 123: 48 124: 48 125: 49 126: 49 127: 49

```

Phase I: 10.000 msec at (level) 128 RCSTs

```

Id: 0, frame: 0 startslot: 3, repeat: 12 amount: 13
Id: 1, frame: 0 startslot: 16, repeat: 13 amount: 14
Id: 2, frame: 0 startslot: 30, repeat: 13 amount: 14
Id: 3, frame: 0 startslot: 44, repeat: 13 amount: 14
Id: 4, frame: 0 startslot: 58, repeat: 14 amount: 15
Id: 5, frame: 0 startslot: 73, repeat: 14 amount: 15
Id: 6, frame: 0 startslot: 88, repeat: 14 amount: 15
Id: 7, frame: 0 startslot: 103, repeat: 14 amount: 15
Id: 8, frame: 0 startslot: 118, repeat: 15 amount: 16
Id: 9, frame: 0 startslot: 134, repeat: 15 amount: 16
Id: 10, frame: 0 startslot: 150, repeat: 15 amount: 16
Id: 11, frame: 0 startslot: 166, repeat: 15 amount: 16
Id: 12, frame: 0 startslot: 182, repeat: 16 amount: 17
Id: 13, frame: 0 startslot: 199, repeat: 16 amount: 17
Id: 14, frame: 0 startslot: 216, repeat: 16 amount: 17
Id: 15, frame: 0 startslot: 233, repeat: 17 amount: 18
Id: 16, frame: 0 startslot: 251, repeat: 17 amount: 18
Id: 17, frame: 0 startslot: 269, repeat: 17 amount: 18
Id: 18, frame: 0 startslot: 287, repeat: 17 amount: 18
Id: 19, frame: 0 startslot: 305, repeat: 18 amount: 19
Id: 20, frame: 0 startslot: 324, repeat: 18 amount: 19
Id: 21, frame: 0 startslot: 343, repeat: 18 amount: 19
Id: 22, frame: 0 startslot: 362, repeat: 19 amount: 20
Id: 23, frame: 0 startslot: 382, repeat: 19 amount: 20
Id: 24, frame: 0 startslot: 402, repeat: 19 amount: 20
Id: 25, frame: 0 startslot: 422, repeat: 19 amount: 20
Id: 26, frame: 0 startslot: 442, repeat: 20 amount: 21
Id: 27, frame: 0 startslot: 463, repeat: 20 amount: 21
Id: 28, frame: 0 startslot: 484, repeat: 20 amount: 21
Id: 29, frame: 0 startslot: 505, repeat: 5 amount: 22
Id: 29, frame: 0 startslot: 514, repeat: 15 amount: 22
Id: 30, frame: 0 startslot: 530, repeat: 21 amount: 22
Id: 31, frame: 0 startslot: 552, repeat: 21 amount: 22
Id: 32, frame: 0 startslot: 574, repeat: 21 amount: 22
Id: 33, frame: 0 startslot: 596, repeat: 22 amount: 23
Id: 34, frame: 0 startslot: 619, repeat: 22 amount: 23
Id: 35, frame: 0 startslot: 642, repeat: 22 amount: 23
Id: 36, frame: 0 startslot: 665, repeat: 23 amount: 24

```

【도 5b】

Id: 37, frame: 0 startslot: 689, repeat: 23 amount: 24
Id: 38, frame: 0 startslot: 713, repeat: 23 amount: 24
Id: 39, frame: 0 startslot: 737, repeat: 23 amount: 24
Id: 40, frame: 0 startslot: 761, repeat: 24 amount: 25
Id: 41, frame: 0 startslot: 786, repeat: 24 amount: 25
Id: 42, frame: 0 startslot: 811, repeat: 24 amount: 25
Id: 43, frame: 0 startslot: 836, repeat: 24 amount: 25
Id: 44, frame: 0 startslot: 861, repeat: 25 amount: 26
Id: 45, frame: 0 startslot: 887, repeat: 25 amount: 26
Id: 46, frame: 0 startslot: 913, repeat: 25 amount: 26
Id: 47, frame: 0 startslot: 939, repeat: 26 amount: 27
Id: 48, frame: 0 startslot: 966, repeat: 26 amount: 27
Id: 49, frame: 0 startslot: 993, repeat: 26 amount: 27
Id: 50, frame: 0 startslot: 1020, repeat: 1 amount: 27
Id: 50, frame: 0 startslot: 1025, repeat: 24 amount: 27
Id: 51, frame: 0 startslot: 1050, repeat: 27 amount: 28
Id: 52, frame: 0 startslot: 1078, repeat: 27 amount: 28
Id: 53, frame: 0 startslot: 1106, repeat: 27 amount: 28
Id: 54, frame: 0 startslot: 1134, repeat: 28 amount: 29
Id: 55, frame: 0 startslot: 1163, repeat: 28 amount: 29
Id: 56, frame: 0 startslot: 1192, repeat: 28 amount: 29
Id: 57, frame: 0 startslot: 1221, repeat: 28 amount: 29
Id: 58, frame: 0 startslot: 1250, repeat: 29 amount: 30
Id: 59, frame: 0 startslot: 1280, repeat: 29 amount: 30
Id: 60, frame: 0 startslot: 1310, repeat: 29 amount: 30
Id: 61, frame: 0 startslot: 1340, repeat: 30 amount: 31
Id: 62, frame: 0 startslot: 1371, repeat: 30 amount: 31
Id: 63, frame: 0 startslot: 1402, repeat: 30 amount: 31
Id: 64, frame: 0 startslot: 1433, repeat: 30 amount: 31
Id: 65, frame: 0 startslot: 1464, repeat: 31 amount: 32
Id: 66, frame: 0 startslot: 1496, repeat: 31 amount: 32
Id: 67, frame: 0 startslot: 1528, repeat: 4 amount: 32
Id: 67, frame: 0 startslot: 1536, repeat: 26 amount: 32
Id: 68, frame: 0 startslot: 1563, repeat: 32 amount: 33
Id: 69, frame: 0 startslot: 1596, repeat: 32 amount: 33
Id: 70, frame: 0 startslot: 1629, repeat: 32 amount: 33
Id: 71, frame: 0 startslot: 1662, repeat: 32 amount: 33
Id: 72, frame: 0 startslot: 1695, repeat: 33 amount: 34
Id: 73, frame: 0 startslot: 1729, repeat: 33 amount: 34
Id: 74, frame: 0 startslot: 1763, repeat: 33 amount: 34
Id: 75, frame: 0 startslot: 1797, repeat: 33 amount: 34
Id: 76, frame: 0 startslot: 1831, repeat: 34 amount: 35
Id: 77, frame: 0 startslot: 1866, repeat: 34 amount: 35
Id: 78, frame: 0 startslot: 1901, repeat: 34 amount: 35
Id: 79, frame: 0 startslot: 1936, repeat: 35 amount: 36
Id: 80, frame: 0 startslot: 1972, repeat: 35 amount: 36
Id: 81, frame: 0 startslot: 2008, repeat: 35 amount: 36
Id: 82, frame: 16 startslot: 3, repeat: 35 amount: 36
Id: 83, frame: 16 startslot: 39, repeat: 36 amount: 37
Id: 84, frame: 16 startslot: 76, repeat: 36 amount: 37
Id: 85, frame: 16 startslot: 113, repeat: 36 amount: 37

【도 5c】

Id: 86, frame: 16 startslot: 150, repeat: 37 amount: 38
Id: 87, frame: 16 startslot: 188, repeat: 37 amount: 38
Id: 88, frame: 16 startslot: 226, repeat: 37 amount: 38
Id: 89, frame: 16 startslot: 264, repeat: 37 amount: 38
Id: 90, frame: 16 startslot: 302, repeat: 38 amount: 39
Id: 91, frame: 16 startslot: 341, repeat: 38 amount: 39
Id: 92, frame: 16 startslot: 380, repeat: 38 amount: 39
Id: 93, frame: 16 startslot: 419, repeat: 39 amount: 40
Id: 94, frame: 16 startslot: 459, repeat: 39 amount: 40
Id: 95, frame: 16 startslot: 499, repeat: 11 amount: 40
Id: 95, frame: 16 startslot: 514, repeat: 27 amount: 40
Id: 96, frame: 16 startslot: 542, repeat: 39 amount: 40
Id: 97, frame: 16 startslot: 582, repeat: 40 amount: 41
Id: 98, frame: 16 startslot: 623, repeat: 40 amount: 41
Id: 99, frame: 16 startslot: 664, repeat: 40 amount: 41
Id: 100, frame: 16 startslot: 705, repeat: 41 amount: 42
Id: 101, frame: 16 startslot: 747, repeat: 41 amount: 42
Id: 102, frame: 16 startslot: 789, repeat: 41 amount: 42
Id: 103, frame: 16 startslot: 831, repeat: 41 amount: 42
Id: 104, frame: 16 startslot: 873, repeat: 42 amount: 43
Id: 105, frame: 16 startslot: 916, repeat: 42 amount: 43
Id: 106, frame: 16 startslot: 959, repeat: 42 amount: 43
Id: 107, frame: 16 startslot: 1002, repeat: 19 amount: 43
Id: 107, frame: 16 startslot: 1025, repeat: 22 amount: 43
Id: 108, frame: 16 startslot: 1048, repeat: 43 amount: 44
Id: 109, frame: 16 startslot: 1092, repeat: 43 amount: 44
Id: 110, frame: 16 startslot: 1136, repeat: 43 amount: 44
Id: 111, frame: 16 startslot: 1180, repeat: 44 amount: 45
Id: 112, frame: 16 startslot: 1225, repeat: 44 amount: 45
Id: 113, frame: 16 startslot: 1270, repeat: 44 amount: 45
Id: 114, frame: 16 startslot: 1315, repeat: 44 amount: 45
Id: 115, frame: 16 startslot: 1360, repeat: 45 amount: 46
Id: 116, frame: 16 startslot: 1406, repeat: 45 amount: 46
Id: 117, frame: 16 startslot: 1452, repeat: 45 amount: 46
Id: 118, frame: 16 startslot: 1498, repeat: 34 amount: 47
Id: 118, frame: 16 startslot: 1536, repeat: 11 amount: 47
Id: 119, frame: 16 startslot: 1548, repeat: 46 amount: 47
Id: 120, frame: 16 startslot: 1595, repeat: 46 amount: 47
Id: 121, frame: 16 startslot: 1642, repeat: 46 amount: 47
Id: 122, frame: 16 startslot: 1689, repeat: 47 amount: 48
Id: 123, frame: 16 startslot: 1737, repeat: 47 amount: 48
Id: 124, frame: 16 startslot: 1785, repeat: 47 amount: 48
Id: 125, frame: 16 startslot: 1833, repeat: 48 amount: 49
Id: 126, frame: 16 startslot: 1882, repeat: 48 amount: 49
Id: 127, frame: 16 startslot: 1931, repeat: 48 amount: 49
Phase II: 30.000 msec at 128 RCSTs